Chinese Journal of Chemical Engineering 22 (2014) 812–819

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJCHE

Process Monitor

Study and Application of Fault Prediction Methods with Improved Reservoir Neural Networks☆

Qunxiong Zhu, Yiwen Jia, Di Peng, Yuan Xu *

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

ARTICLE INFO

Article history:
Received 15 May 2013
Received in revised form 12 October 2013
Accepted 24 December 2013
Available online 23 June 2014

Keywords:
Fault prediction
Time series
Reservoir neural networks
Tennessee Eastman process

ABSTRACT

Time-series prediction is one of the major methodologies used for fault prediction. The methods based on recurrent neural networks have been widely used in time-series prediction for their remarkable non-liner mapping ability. As a new recurrent neural network, reservoir neural network can effectively process the time-series prediction. However, the ill-posedness problem of reservoir neural networks has seriously restricted the generalization performance. In this paper, a fault prediction algorithm based on time-series is proposed using improved reservoir neural networks. The basic idea is taking structure risk into consideration, that is, the cost function involves not only the experience risk factor but also the structure risk factor. Thus a regulation coefficient is introduced to calculate the output weight of the reservoir neural network. As a result, the amplitude of output weight is effectively controlled and the ill-posedness problem is solved. Because the training speed of ordinary reservoir networks is naturally fast, the improved reservoir networks for time-series prediction are good in speed and generalization ability. Experiments on Mackey–Glass and sunspot time series prediction prove the effectiveness of the algorithm. The proposed algorithm is applied to TE process fault prediction. We first forecast some time-series obtained from TE and then predict the fault type adopting the static reservoirs with the predicted data. The final prediction correct rate reaches 81%.

1. Introduction

Fault detection and diagnosis have been studied for almost four decades and become a significant part in control theory. The requirements for system reliability and safety are increasing. It is crucial to know the failure information before a fault. As a result, fault prediction has attracted much attention.

The key of fault prediction is to forecast the future state of a system, so fault prediction can be transformed to time-series prediction. The existing methods of time-series prediction can be classified into three categories. The first method is based on classical time series analysis, consisting of ARMA model and ARIMA model [1]. The second method is based on gray model [2,3] and the last one is based on neural networks [4–7]. Among these methods, the neural network method has been studied deeply and applied to time-series extensively for its remarkable non-linear mapping ability. In neural networks and machine learning communities, several types of neural network model are applied to time-series prediction such as the standard multilayer perceptions [8], radial basis function neural networks [9–11], and generalized regression neural networks [12]. In addition, recurrent neural networks [13] including nonlinear autoregressive network [14], extreme learning machine networks [15], and recurrent predictor neural networks [16] are also studied for nonlinear time-series prediction.

There are some limitations when applying neural networks to applications. For example, the performance is not good when the forward neural network is applied to time-series prediction. Although recurrent neural networks can solve the problems related to time-series, it has many disadvantages such as large calculation, slow convergence rate and difficulty in determining the number of hidden neuron. Moreover, there are fading memories, which may make error gradient missing or distorted.

To solve these problems, Jaeger and Maass proposed echo state networks (ESNs) [17] and liquid state machine [18], respectively. Although these two methods have different angles, their essence is the improvement of traditional recurrent neural networks. Verstraeten *et al.* have demonstrated that the two methods are the same in essence and named it reservoir computing [19]. Since the report of reservoir computing on Science journal in 2004 [20], it has drawn a number of researchers' attention around the world. Beside time-series prediction [21,22], reservoir computing is extended to pattern classification [23], voice recognition [24], image processing [25] and so on.

However, there are some problems in reservoir networks. In many situations, the coefficient matrix for calculating the output weight is

☆ Supported by the National Natural Science Foundation of China (61074153).
* Corresponding author.
E-mail address: xuyuan@mail.buct.edu.cn (Y. Xu).

http://dx.doi.org/10.1016/j.cjche.2014.05.016
1004-9541/

morbidly. To be specific, singular values distribute continuously with no obvious jump. The maximum singular value and the minimum one differ significantly. Consequently, the output weight is extraordinary large especially in the high-dimension reservoir networks. On the other hand, the conventional way to control the output weight is choosing the reservoirs with the dimension as low as possible, but the low-dimension reservoir networks cannot bring good generalization ability.

In this paper, we first study the traditional structure of reservoir networks and analyze its ill-posedness problem. According to the analysis, the structure risk is taken into consideration. A formula is obtained to calculate the output weight with minimizing the loss function. This method involves a regulating coefficient that can control the amplitude of the output weight. The ill-posedness problem is solved in this way. Experiments of two benchmark problems are used to verify the effectiveness of the improved method. A fault prediction algorithm based on the improved reservoir neural networks is proposed and applied to TE process. Six time-series data consisting of 2 variables from 3 faults are predicted. In the classification stage, we take advantage of static reservoir networks to predict the type of the faults.

2. Reservoir Computing

2.1. Structure of reservoir network

The architecture of traditional reservoirs [17] is shown in Fig. 1. Some terminologies must be fixed first. We consider discrete-time neural networks with K input units, N internal network units and L output units. Activations of input units at time step n are $\boldsymbol{u}(n) = (u_1(n), \ldots, u_M(n))$, those of internal units are $\boldsymbol{x}(n) = (x_1(n), \ldots, x_N(n))$, and those of output units are $\boldsymbol{y}(n) = (y_1(n), \ldots, y_L(n))$. Real-valued connection weights are collected in a $N \times K$ weight matrix $\boldsymbol{W}^{\text{in}} = (w_{ij}^{\text{in}})$ for the input weights, in an $N \times N$ matrix $\boldsymbol{W} = (w_{ij})$ for the internal connections, in an $L \times (K + N + L)$ matrix $\boldsymbol{W}^{\text{out}} = (w_{ij}^{\text{out}})$ for the connection to the output units, and in a $N \times L$ matrix $\boldsymbol{W}^{\text{back}} = (w_{ij}^{\text{back}})$ for the connections that project back from the output to the internal units.

2.2. Mathematical model

In most cases the output has little effect on internal unit, so we will not study parameter $\boldsymbol{W}^{\text{back}}$. The equations of the reservoir networks [17] can be written as

$$x(k) = f\left(\boldsymbol{W} \cdot x(k-1) + \boldsymbol{W}^{\text{in}} \cdot u(k) + b_x\right)$$
$$y(k) = \boldsymbol{W}^{\text{out}} x(k) + b. \tag{1}$$

Considering Eq. (1), we assume that the internal state variables $\boldsymbol{x}$ have N dimensions, input variables $\boldsymbol{u}$ have M dimensions, and output variables $\boldsymbol{y}$ have L dimensions. To simplify the expressions, we consider bias variables as the connection weight of output fixed value of 1. Thus b_x and b can be merged into matrices $\boldsymbol{W}^{\text{in}}$ and $\boldsymbol{W}^{\text{out}}$. The active function f can be spiking neurons, threshold logic neurons, sigmoid neurons, linear neurons and so on. In this paper, sigmoid function is taken. We first initialize the networks: $\boldsymbol{W}$ and $\boldsymbol{W}^{\text{in}}$ are generated randomly and remain unchanged during the calculation, and the original state of internal unit is zero, that is, $x(0) = 0$. The input and output of training samples are $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$, respectively. Thus we can calculate $\boldsymbol{W}^{\text{out}}$ with Eq. (1).

Some important points on reservoir networks are presented as follows.

First, the dimension of internal state units $\boldsymbol{x}$ is very high, up to hundred even thousand, while it is relatively lower in the traditional recurrent neural networks.

Second, weight matrices $\boldsymbol{W}^{\text{in}}$ and $\boldsymbol{W}$ are randomly generated and remain unchanged during all training processes.

Last, as one of the measures maintaining the dynamic characteristics of reservoirs, the connection weight matrix of internal state is sparse to the point of 2%–5%, different from most traditional recurrent neural networks, which always keep dense connection.

3. Improved Method

In this section, the ill-posedness problems of reservoirs networks are discussed and the improved method solving the ill-posedness problems will be given.

3.1. Training

In order to facilitate the study, we redraw a new picture of the network structure in Fig. 2, which is essentially the same as the structure in Fig. 1, with the same neuron of reservoir in different moments. $\boldsymbol{W}^{\text{in}}$ is the connection weight between the input layer and the reservoir, $\boldsymbol{W}$ is the interval connection weight, and $\boldsymbol{W}^{\text{out}}$ is the connection weight between the output layer and the reservoir.

Training the reservoir networks can be summarized as determining the connection weight matrix $\boldsymbol{W}^{\text{out}}$ between the output layer and the dynamic reservoir layer. The following are the detailed steps of building and training a reservoir network.

Step 1 Set the parameters of reservoir networks. Set the number of internal units of reservoirs (N), the sparsity, and the spectral radius of the connection weight matrix of internal state. Initialize the reservoir network. The spectral radius of the connection weight matrix of internal state is always between 0 and 1, but it is not a necessary condition. Sometimes the spectral radius greater than 1 can give better prediction performance.

Step 2 Calculate the interval state. Normalize the input sample and stimulate the internal state of reservoirs using the normalized

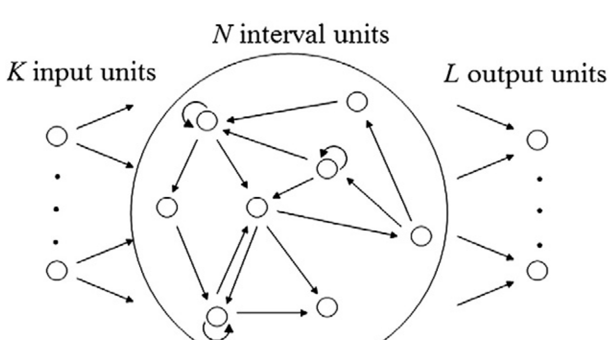

Fig. 1. The basic structure of reservoir networks.

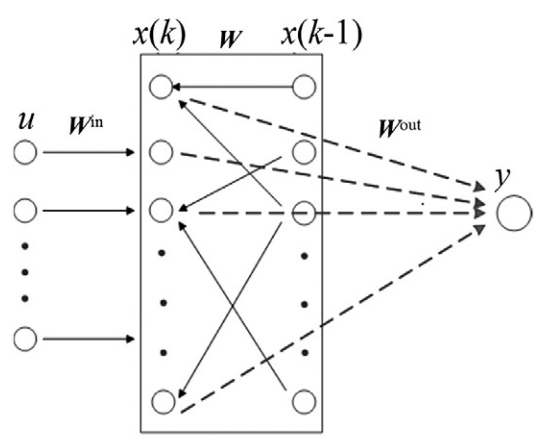

Fig. 2. The simplified model of reservoir.